2014

Listening to botnet communication channels to protect information systems

Brian Cusack
Auckland University of Technology, brian.cusack@aut.ac.nz

Sultan Almutairi
Auckland University of Technology, sultan.almutairi@aut.ac.nz

DOI: 10.4225/75/57b3df16fb87b

12th Australian Digital Forensics Conference. Held on the 1-3 December, 2014 at Edith Cowan University, Joondalup Campus, Perth, Western Australia.

LISTENING TO BOTNET COMMUNICATION CHANNELS TO PROTECT INFORMATION SYSTEMS

Brian Cusack; Sultan Almutairi
Auckland University of Technology, Auckland, New Zealand
brian.cusack@aut.ac.nz sultan.almutairi@aut.ac.nz

ABSTRACT

Botnets are a weapon of choice for people who wish to exploit information systems for economic advantage. A large percentage of high value commercial targets such as banking transaction systems and human customers are web connected so that access is gained through Internet services. A Botnet is designed to maximise the possibility of an economic success through the low cost of attacks and the high number that may be attempted in any small time unit. In this paper we report exploratory research into the communications of Botnets. The research question was: How do Botnets talk with the command and control channels? The research method is to catch binaries in a low interaction honey pot and then to provide a secure test bed in which the binaries can demonstrate the actions of malicious activity. One of the actions performed by a binary is communication with the Bot master and this action is the focus of our study. We also provide a feedback loop in which suggestions are made to protect an Information System and the users.

Keywords: Botnets, Cybercrime, Covert, Communications, Disruption

INTRODUCTION

Botnets are responsible for a large percentage of damages and criminal activity on the Internet. The business model is simple. Someone (the master or herder) sets up a network of control over many computers (Bots) and steals the computing and communication system resources. The stolen capabilities are then on-sold to willing buyers who make a living from spamming, theft of personal identities, extortion, DDOS attacks and so on. It is a simple economic formula that delivers the promise of high financial gains to those involved and at the expense of legitimate Internet service users. In previous publications we have reported the steps of forensic investigation for such events from the victim's perspective. In this paper the research is escalated to where the communications between the Bot Master and the Bots is investigated. These communications are for initial connection, command and control instructions and for maintenance activities. One consequence of such research is the potential disruption of the covert channels and another is a better understanding of the overall Botnet. The implications are for protecting legitimate Internet users and for finding measured responses to economic harm.

The evolution of Botnet attacks from push to pull has made research more difficult. In the investigation of a Botnet using a traditional method such as a push-based model, researchers might locate the attack vector by finding vulnerabilities in the system with penetration testing and by reconstructing the event. However, to find the initial phase of an attack in push-based Botnet methods, researchers must evaluate the various possibilities of how the Botnet malwares were distributed and in particular the actions of the end users who may have permitted the access (Schiller, Binkley, Evron, Willems, Bradley, Harley, 2007). The switch from push-based research where the malwares remotely intrude a system through security flaws, to a pull-based model where the unwitting host performs an action such as a download or a mouse click considerably extends the scope of research and the potential costs (Provos, Mavrommatis, Rajab, Monrose, 2008). One of the propagation techniques in the pull model is using various social engineering techniques. For example, attackers gather visitors of a website with phishing methods, and allow the visitors to accidentally download the malware. Another technique involves exploitation of various browser vulnerabilities. In this case, visitors come to automatically download malware and run it without their knowledge. Using the techniques, the number of victims can be easily increased without any traditional security barriers because conventional protection mechanisms cannot prevent the victim actions (Chiang, Lloyd, 2007). A researcher is hence challenged to use a multiplicity of approaches and methods from both qualitative and qualitative ways to research Botnet activities.

The aim of this research is to observe the behaviour of the binaries that give the Botnet the capability to expand and create zombies of other systems. The communication between the Bot and the Bot master and the Bot and other Bots is of particular interest when the communications are observed in relation to Bot actions. However around 90% of malware binaries employ analysis-resistance techniques (Semantic Security Response, 2010) and hence the work is expected to be challenging (Bailey, Cooke, Jahanian, Xu, Karir, 2009). The remainder of this

paper is structured to review previous literature, report our findings and to discuss the possibilities of exploiting the knowledge to protect Internet users from harm.

PREVIOUS LITERATURE

A Botnet is a collection of computers or a large network of compromised computers (Ullah, Khan, Aboalsamh, 2013). A Bot refers to malicious software that runs on an infected computer and gives control to the attacker (Rajab, Zarfoss, Monrose, Terzis, 2006). A Bot is also known as a virus of viruses (Schiller, et al., 2007). The attacker controls Bots by using a C&C command channel for the exchange of instructions for actions (Correia, Rocha, Nogueira, Salvador, 2012). The attacker usually uses one or more servers in order to allow continuous communication and to off load stolen information (Zahid, Belmekki, Mezrioui, 2012). The command received through the C&C channel is executed autonomously and automatically without the end user's consent. The Botnet is also known as zombies because the malicious intent is hidden until activated by an instruction (Choo, 2007). Also the attacker who controls the C&C server is called the Bot master or the master (Rajab et al., 2006). The primary difference between the Bot clients and viruses or worms is that Bot clients are able to take an action autonomously and execute the given commands in a coordinated manner (Schiller et al., 2007). Bot clients have the ability to perform their actions when attackers are not logged into the target machine. For this reason, a Botnet can be classified by the C&C which are usually IRC Internet, P2P or HTTP (Chiang, Lloyd, 2007). When a Bot discovers or receives a new opportunity on a victim system, it can automatically install a specific module to distribute the malware. It means that defeating one component of a Botnet is not enough to ensure that the entire system is cleaned up. Also the Bots utilize a number of techniques to increase continuity and stability depending on the situation of a specific system targeted (Hoagland, Ramzan, Satish, 2008). In cases where authorities disrupt a C&C server at a certain IP address, the Bot master can easily set up another C&C server instantly with the same name at a different IP address.

Botnet research usually starts with the active collecting of samples or the passive detection of Bot behaviours (Mell, Kent, Nusabaum, NIST). Honeypots have been widely used as an information system resource whose value lies in unauthorized or illicit use of that resource (The Honeypot Project, 2007). Baecher et al. (2006) argue that the collecting and analysing of malware samples provides a better defence against the existing threats and also against potential events. In particular, statistical information generated from the large scale samples can be useful to learn about the patterns, trends, and types of attack. The honeypot technologies have been recognised as good sample providers in several Botnet research studies (Cooke, Jahanian, McPherson, 2005; Freiling, Holz, Wicherski, 2005). Detecting Botnets is another approach using passive network traffic monitoring and analysis. These techniques have been useful to identify the existence of Botnets by detection of behaviours associated with groups of compromised machines within a monitored network. Gu et al. (2008) conducted research in which they assumed that Bots within the same Botnet could be characterized by their protocols such as network communication traffic and malicious activities. Based on this assumption, the researchers categorised Bots by using IRC protocol and executed a large number of Bot samples obtained by this categorising. These efforts enabled them to identify the first level of IRC servers and then infiltrate the corresponding IRC communication channels to snoop on the Botnets (Feily, Shahrestani, Ramadas, 2009). The challenges for researchers are noted in several reports where Stealth and deception techniques have been observed changing continuously to avoid detection and analysis. A Botnet can change its C&C server address frequently during its lifetime by using fast-flux service networks (Bacher, Holz, Koetter, Wicherski, 2008; Holz, Gorecki, Rieck, Freiling, 2008). Similarly the usual techniques for detecting the existence of malware is based on the signatures of a binary file such as byte sequences and strings (Tabish, Shafig, Faroog, 2009). The signature based malware detection can be easily defeated by packer and binary code obfuscation techniques (Stepan, 2006).

TEST SET UP

The test set-up was informed by the literature reviewed and was designed into two parts. One part to trap the Botnet binaries in a low interaction honeypot and the other to release these in a controlled environment independently to study the communication behaviours. The purpose was to understand the scope of communications between a Bot master and the Bot in relation to the actions of the Bot. The attempt was to assess the extent to which actions are related to communications, the origins and the destinations of communications, and the different types of communication. A Bot Master is responsible for all the social communications between the Bots as well as the communication between the Bots and the server. A communication suite has three functionalities called a botworker, a botupdater and a C&C engine. In the first functionality, a Bot Master builds and maintains the bots to be able to infect different types of machine as well as the communication between them. Botupdater is for communications that update the bots with a new software or a new command. The botupdater functionality also operates when a C&C channel has been disrupted. It can

update the Bots with a new C&C engine. The C&C engine works like a warehouse that forwards the messages from the Bot Master to the bots and authenticates the channel contents. Consequently we were alert to the types of communication to look for and some of the complexities that may be encountered. Other key metrics noted were:

- The Response Time should be fast to be considered as a command and that is speed of 100ms for incoming packets and 3 seconds for outgoing packets.
- The size of the communication sessions should be small which reflects the small size of the command which is less than 1KB for hiding detection.
- The Time Interval which is the time between receiving the command the time for the application to be launched in the infected host may vary.
- The Session Count and the Destination Count and the Average duration Count should be low again for obfuscation purposes.

The overview of the communications for both incoming and outgoing traffic can be progressively recorded and the Bot communications isolated. In the test setup the binaries will be held in isolation from a live network and the attributes of the Internet simulated. The presentation techniques of the communications are graph visualization of all traffic, scatter plots of time intervals and parallel histograms for time series. These visualizations may cover all network traffic but the Bot C&C related traffic are to be reported. Figure 1 shows the five phases of the research from the acquisition of the binaries through to the report of the Bot communication behaviours.

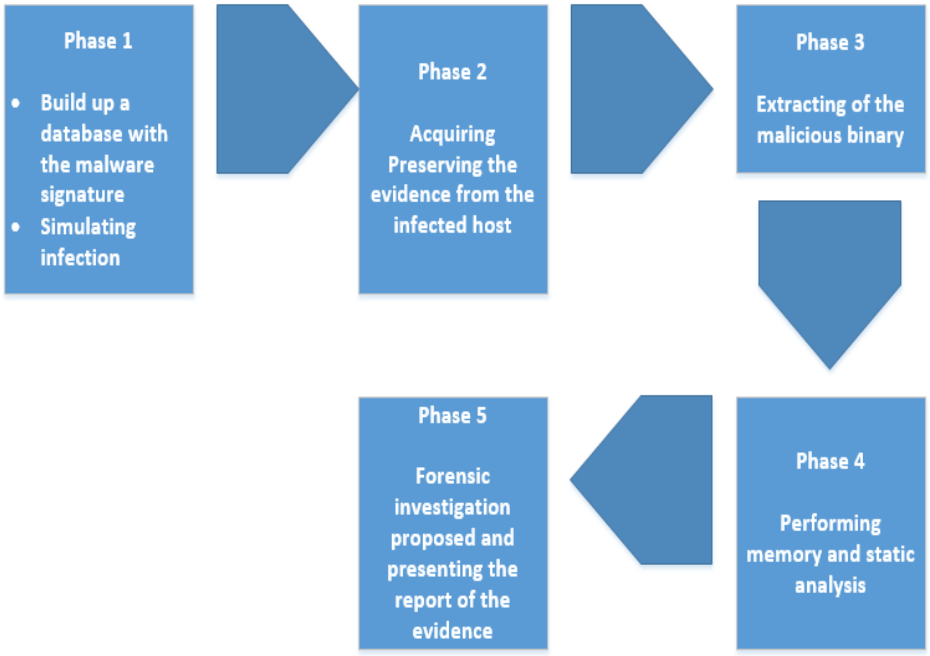

Figure 1. The Research Phases

THE RESULTS

The honeypot reported after 22 days the download of 60 unique malware binaries. These binaries were then brought into the secure test setup and provided with environments in which there was a good chance they could demonstrate the behaviours. Some were more active than others and each exhibited different behaviours including requesting different numbers of external files, periodic behaviours and communication types. The observations performed showed the communication between the infected host and through the C&C channel.

The researcher was able to capture the domain names and the IP addresses requested by the infected host and to monitor the communication behaviours. The result shows that Botnet C&C channels can be monitored and the related behaviours observed. The following figures provide evidence of the different types of communications between an IRC Bot and a Bot master. [Note: Actual IP addresses have been damaged to preserve privacy; Dynamic addresses were correct at the time of the action but may change and have various unrelated ownership at any other point in time.]

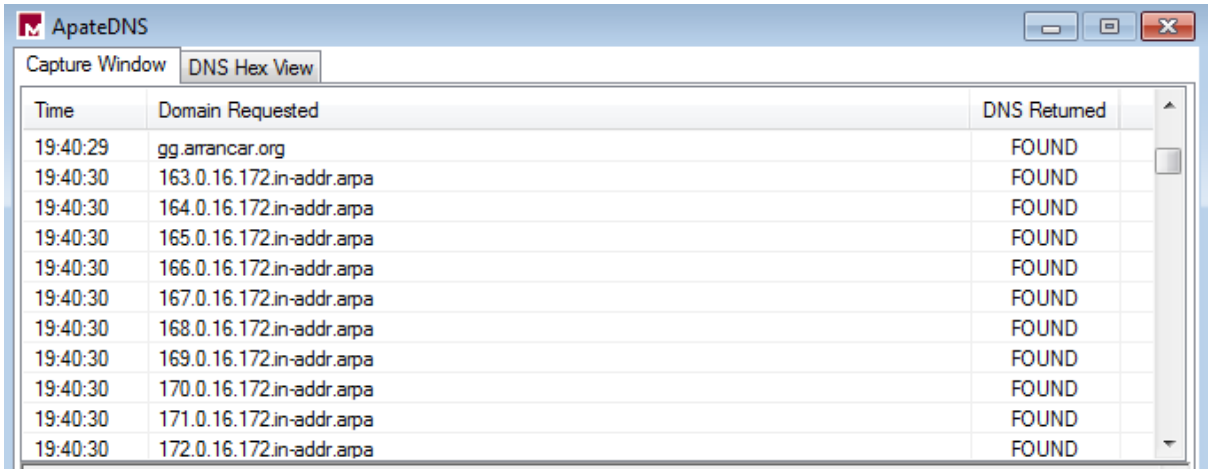

Figure 2. Bot C&C Communication

Figure 2 shows the domain names and the IP addresses requested after the infection of a Bot by the IRC bot. The Wireshark tool was able also to capture these domain names and IP addresses as they were communicated. The Bot performed these actions autonomously.

```
 75 Standard query 0xed0a  A gg.arrancar.org
135 Standard query response 0xed0a  A 128.111.73.201
```

Figure 3. Bot Query Communications

In figure 3 a target gg.arrancar.org is requested by the Bot. The internet connection was enabled for a short period of time and more than 200 domain name IP addresses were requested.

Figure 4. Malware Identification

In figure 4 the Bot was identified by hash value in a security database so that expected behaviours could be predicted. Also we accessed other reports to find expected behaviours from the different perspectives of different analysts.

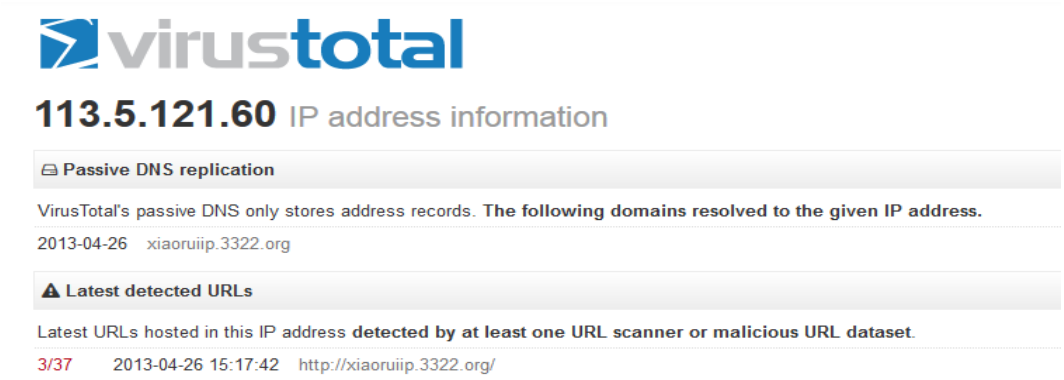

Figure 5. C&C Host Address

The researcher performed a google search for the domain name gg.arrancar.org and the figure 5 from the Microsoft website shows that this domain is being used as malicious host to control a Botnet. Figure 5 also shows an alias of the gg.arrancar.org indicating obfuscation in the C&C messaging.

```
<stateVariable sendEvents="no">
<name>PortMappingLeaseDuration</name>
<dataType>ui4</dataType>
</stateVariable>
<stateVariable sendEvents="no">
<name>RemoteHost</name>
<dataType>string</dataType>
</stateVariable>
<stateVariable sendEvents="no">
<name>ExternalPort</name>
<dataType>ui2</dataType>
</stateVariable>
<stateVariable sendEvents="no">
<name>InternalPort</name>
<dataType>ui2</dataType>
</stateVariable>
<stateVariable sendEvents="no">
<name>PortMappingProtocol</name>
<dataType>string</dataType>
<allowedValueList>
<allowedValue>TCP</allowedValue>
<allowedValue>UDP</allowedValue>
</allowedValueList>
</stateVariable>
<stateVariable sendEvents="no">
<name>InternalClient</name>
<dataType>string</dataType>
</stateVariable>
<stateVariable sendEvents="no">
<name>PortMappingDescription</name>
<dataType>string</dataType>
</stateVariable>
<stateVariable sendEvents="no">
<name>UserName</name>
<dataType>string</dataType>
</stateVariable>
<stateVariable sendEvents="no">
<name>Password</name>
<dataType>string</dataType>
</stateVariable>
<stateVariable sendEvents="no">
<name>PPPEncryptionProtocol</name>
<dataType>string</dataType>
</stateVariable>
<stateVariable sendEvents="no">
<name>PPPCompressionProtocol</name>
<dataType>string</dataType>
</stateVariable>
<stateVariable sendEvents="no">
<name>PPPAuthenticationProtocol</name>
<dataType>string</dataType>
</stateVariable>
<stateVariable sendEvents="no">
<name>AutoDisconnectTime</name>
<dataType>ui4</dataType>
</stateVariable>
<stateVariable sendEvents="no">
<name>IdleDisconnectTime</name>
<dataType>ui4</dataType>
</stateVariable>
<stateVariable sendEvents="yes">
<name>X_Name</name>
<dataType>string</dataType>
</stateVariable>
</serviceStateTable>
</scpd>
```

Figure 6. Bot Authentication Communication

Figure 7. Host Authentication Communication

Figures 6 and 7 show direct communication between the Bot and the Bot master where authentication is required by the Bot master before the Bot is acknowledged as one of the Botnet (figure 7). In figure 6 the Bot has contacted the Bot master and is requesting authentication instructions and for new instructions to be sent. The captured traffic were sent from the infected host through the C&C channel. Here we observed the link that was unique to the infected host to the C&C channel which is http://XXX.168.1.1:80. The http://XXX.168.1.1:80 is the IP address of the infected host that is unique to it with the communication being send using port 80. The C&C channel would be able to identify each host by its IP address and its origin. The research shows that the IRC bot uses the TCP traffic in a plain text, and that it checks the status of the infected host to see if it is still alive and doing the botnet army business. It uses the Internet control message protocol (ICMP) to check the status of the IRC Bot (figure 8). The instructions in the C&C channel of the IRC Bot used TCP for the communication between the infected host and in the C&C channel. The Wireshark tool was able to capture the communications. The communications that were captured by the Wireshark include GetUserName, UserName, Password, NewUpstreamMaxBitRate, NewDownstreamMaxBitRate, ConfigureConnection, NewUserName, NewPassword, InternalPort, RemoteHost, ExternalIPAddress, SetDefaultConnectionService, NewDefaultConnectionService and other information that been sent from the infected host to the C&C channel. All of these communications were able to be intercepted. In figure 8 the communication between the Bot master and the Bot and vice versa is shown.

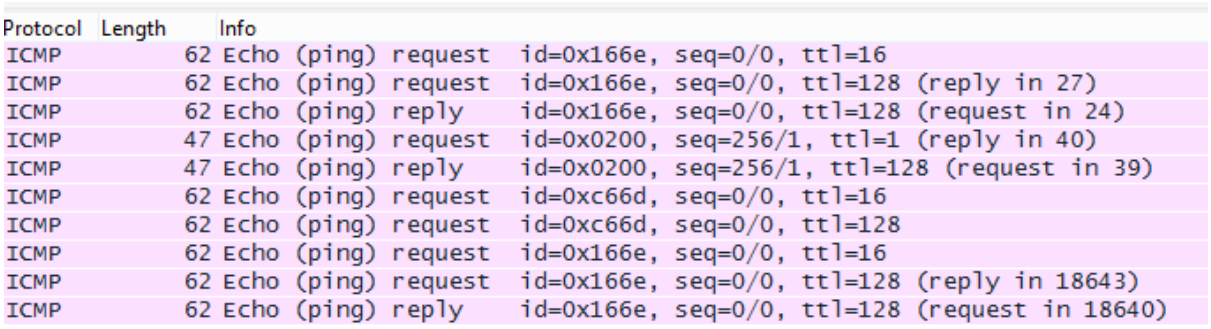

Figure 8. Status Communication

Therefore, it is believed that the IRC bot send the instructions including the attack instructions in a plain text format which is one of the characteristics of the IRC bot. However, many Botnet masters use the IRC Bot because it is easy to set up as well as many of them have many years of experience using this type of Botnet. The malware collection reports show that the IRC Bot is still one of the common Botnets in the internet community. Most Botnets use many C&C channel hosts to be able to communicate with many Bots. Hence it is not only complicated to track the active channel down but also near impossible to destroy. The Anubis sandbox shows that one of the IRC Bots requested more than 140 hosts. The live monitoring of the infected host showed that more than 200 hosts were requested. In addition, the registry of the infected hosts were changing continuously during the research and the live monitoring of the infected host when it was connected to the internet server.

DISCUSSION

The research question asked; How do Botnets talk with the command and control channels? The results of our exploratory observations show that the communication is principally between the Bot master and the individual Bots. The command and control channels are subsequently varied on a regular basis so as to hide the location and identity of the Bot master. The most common and active Bots we had in the controlled laboratory environment were IRC Bots. These Bots have characteristics that are common to older and more established architectures. If we had put more time into some of the other exotic Bots that were few in number and others that were inactive or demonstrated few behaviours our results may have been different. The malware analysis

tools were used to be able to analyse the identity and expected behaviours of the Bots. They provided useful information about expectations so we could look for these and also the observation that most of the binaries have been written in C++ language. This means that the developers of the Botnet are accomplished programmers with strong systems knowledge. In addition sniffer tools were installed in the infected host to capture communications. It showed which Bots are pre-programmed to perform in the infected host once the host is infected and disclosed the list of actions that the bot is familiar with. The sniffer tools caught messages being transferred by the Bot to the suspicious C&C channel.

The research design was made to be similar to a real Botnet event. The only controls enforced were security controls to prevent contact from the test setup to other networks. The matter that could not be controlled was the Bot behaviour. Some Bots are aware they have been caught in a honeypot and they change their behaviour accordingly. Others go into a sleeping mode so as to not disclose the behaviours. The IRC Bots however appeared more simplistic and open to observation. They for example communicated in plain text and enjoyed the research environment. The operating system (OS) of the infected host used in this research were Windows XP and Windows 7. The reason for choosing the Windows OS in particular is that the majority of the malware are targeting Windows OS and we felt the Bots would be more comfortable there. Both versions of Windows OS were used as Virtual machines through VMware Workstation (VM). The host were infected with many IRC Bots and the behaviour of them were mostly similar. However, there were some differences such as the number of domains and IP addresses that have been connected to, and also two of the IRC Bots forced the Windows 7 to restart as well as two others required the Bot to run the OS with administrator rights. The Widows operation system (Windows XP and Windows 7) were not turned off after the infection of the host. The OS were examined including the physical memory of the infected host. The reason for not turning off the machines is that the physical memory of the infected host will be deleted at the time of the infection if the host is turned off. The image of the registry files was taken by Regshot to be able to determine the changes that have been done to the infected host. Then the infection process of the host was monitored by the malware tools and other tools that are provided by Windows Corporation. The Sniffer tools monitored the communication traffic of the infected host using Wireshark and ApateDNS. These processes occurred in the back ground without the knowledge of the owner of the machine.

In our previous publications we have assessed the complexity of Botnet investigations on a scale of low cost and complexity Level 1 to high cost and complexity Level 5. The listening to C&C communications is at Level 4 and the destruction of the Botnet at Level 5. The tracking of the Bot master is at a different level from the research we undertook and has a different disruptive strategy. Botnets usually involve international incidents and cross border jurisdictional matters (See Khan et al., 2014 for example for Level 5 investigation techniques). As a consequence the policing of these matters is difficult. One of the main issues in stopping this type of incident is that there are still countries that do not have a cybercrime law and laws that criminalise the activities hosted by Botnet attacks. Hence an attacker can operate from such a jurisdiction and into others without fear of accountability. There are many examples of botnet events where the botnet-master prosecution would require an international effort such as Aramco Oil Company that is located in Saudi Arabia that was attacked by a Botnet-masters group. The internal investigation of the incident showed that the damages caused by the event were severe with more than 30,000 infected machines and the originators of the attack came from four countries in four different continents. Another example of the Cross Border Issues is when the Mariposa botnet masters managed to steal sensitive information from 800,000 users across 190 countries. These two examples show just how the international effort should be gathered to be able to stop this type of cybercrime from destroying the internet environment for legitimate users. The joint international effort was able to arrest the three Mariposa botnet masters in Spain, however, our research suggests that intelligence can be gained at a Level 4 investigation that may assist Level 5 actions. Principally a Level 4 investigation can provide intelligence and also inform strategy that disrupts the Bot and Bot master communication channels. Consequently Level 4 research has a practical application and may assist Information System protection when Level 5 actions are impossible, or made difficult by for example cross border matters.

CONCLUSION

The contribution of our research is to demonstrate the ease with which many Bots particularly of the IRC type may be listened to. The implications are for developing disruptive strategies. Botnets remain a challenge for the legitimate users of the Internet and their freedom from economic harm. We have demonstrated ways in which the problem can be approached when it is not possible to find the Bot master. Cross-border issues are one of the challenges hindering international co-operation to stop this type of crime. As a consequence listening to the C&C channels of Botnets can be the starting point for disrupting the activity. Better protection for an Information System can be gained by educating users to manage excesses in their own online behaviour and to

adopt a defensive position to online trickery and scams. In addition proactive firewall defences and updated and strong anti-virus software are a deterrent to push attacks.

REFERENCES

Bächer, P., Holz, T., Kötter, M., Wicherski, G. (2008). Know your Enemy: Tracking Botnets. Retrieved Oct 01, 2013, from http://www.honeynet.org/papers/bots/

Baecher, P., Koetter, M., Holz, T., Dornseif, M., Freiling, F. (2006). The Nepenthes Platform: An Efficient Approach to Collect Malware. Proceedings of the 9th International Symposium on Recent Advances in Intrusion Detection (RAID 2006), Hamburg, Germany. doi:10.1007/11856214_9

Bailey, M., Cooke, E., Jahanian, F., Xu, Y., Karir, M. (2009). A Survey of Botnet Technology and Defenses. Proceedings of the 2009 Cybersecurity Applications & Technology Conference for Homeland Security. Doi:10.1109/CATCH.2009.40

Chiang, K., & Lloyd, L. (2007). A Case Study of the Rustock Rootkit and Spam Bot. Proceedings of the First Workshop on Hot Topics in Understanding Botnets, Cambridge, MA. Retrieved from http://www.usenix.org/event/hotbots07/tech/full_papers/chiang/chiang.pdf

Choo, K. (2007). Zombies and botnets. Canberra: Australian Institute of Criminology. Retrieved from http://www.aic.gov.au/en/publications/current%20series/tandi/321-340/tandi333.aspx

Cooke, E., Jahanian, F., McPherson, D. (2005). The Zombie roundup: understanding, detecting, and disrupting botnets. Proceedings of the Steps to Reducing Unwanted Traffic on the Internet (SRUTI '05), Cambridge, MA.

Correia, P., Rocha, E., Nogueira, A., Salvador, P. (2012). Statistical Characterization of the Botnets C&C Traffic. *Procedia Technology, 1*, 158-166.

Daswani, N., Stoppelman, M. (2007). The Anatomy of Clickbot.A. Proceedings of the First Workshop on Hot Topics in Understanding Botnets, Cambridge, MA.

Feily, M., Shahrestani, A., Ramadass, S. (2009). A Survey of Botnet and Botnet Detection. Proceedings of the Emerging Security Information, Systems and Technologies Conference, 2009. SECURWARE '09.

Freiling, F. C., Holz, T., Wicherski, G. (2005). Botnet Tracking: Exploring a Root-Cause Methodology to Prevent Distributed Denial-of-Service Attacks. *Computer Security – ESORICS 2005* (pp. 319-335). Retrieved from http://dx.doi.org/10.1007/11555827_19

Gu, G., Perdisci, R., Zhang, J., Lee, W. (2008). Botminer: Clustering analysis of network traffic for protocol- and structure-independent botnet detection. Proceedings of the 17th USENIX Security Symposium, San Jose, CA.

Hoagland, J., Ramzan, Z., Satish, S. (2008). Bot Networks. In M. Jakobsson & Z. Ramzan (Eds.), *Crimeware: Understanding New Attacks and Defenses* (pp. 183-227): Addison-Wesley Professional.

Holz, T., Gorecki, C., Rieck, K., Freiling, F. C. (2008). Measuring and Detecting Fast-Flux Service Networks. Proceedings of the 15th Annual Network and Distributed System Security Symposium (NDSS' 08), San Diego, CA.

Khan, R., Mizan, M., Hasan, R., Sprague, A. (2014). Hot Zone Identification: Analyzing Effects of Data Sampling on SPAM Clustering. The Proceedings of Conference on Digital Forensics, Security and Law, Richmond, Virginia, May 28-29.

Mell, P., Kent, K., Nusabaum, J. NIST. Guide to Malware Incident Prevention and Handling. Special Publication 800-83. National Institute of Standards and Technology. Washington DC, USA.

Provataki, A., Katos, V. (2013). Differential Malware Forensics. *Digital Investigation*, 10, 311-322.

Provos, N., Mavrommatis, P., Rajab, M. A., Monrose, F. (2008). *All your iFRAMEs point to Us*, San Jose, CA. : Wiley.

Rajab, M. A., Zarfoss, J., Monrose, F., Terzis, A. (2006). A multifaceted approach to understanding the botnet phenomenon. Proceedings of the 6th ACM SIGCOMM conference on Internet measurement, Rio de Janeriro, Brazil.

Schiller, C., Binkley, J., Evron, G., Willems, C., Bradley, T., Harley, D. (2007). *Botnets: The Killer Web App*. Burlington, MA: Syngress.

Stepan, A. (2006). Improving proactive detection of packed malware. Retrieved 28 Sebtember, 2012, from http://www.virusbtn.com/virusbulletin/archive/2006/03/vb200603-packed

Symantec Security Response. (2010). Symantec Global Internet Security Threat Report: Trends for 2009 (Technical Report): Symantec Corporation. Retrieved from http://eval.symantec.com/mktginfo/enterprise/white_papers/b-whitepaper_internet_security_threat_report_xv_04-2010.en-us.pdf

Tabish, S., Shafiq, M., Farooq, M. (2009). Malware detection using statistical analysis of byte-level file content. Retrieved October 2013 from http://ro.ecu.edu.au/cgi/viewcontent.cgi?

Ullah, I., Khan, N., Aboalsamh, H. (2013). Survey on BOTNET: Its Architecture, Detection, Prevention and Mitigation. *IEEE Transactions on Forensics and Security*, *660-665*.

The Honeynet Project. (2007). Know Your Enemy: Fast-Flux Service Networks. Retrieved 15 September, 2012, from http://www.honeynet.org/papers/ff

Zahid, M., Belmekki, A., & Mezrioui, A. (2012). A new architecture for detecting DDoS/Brute force attack and destroying the botnet behind. *IEEE Transactions in Forensics and Security*, 1-5.